Title of the Invention

Automated Design and Execution of Experiments with Integrated Model Creation for Semiconductor Manufacturing Tools

Inventor: Alexander T. Schwarm

Related Application

This application claims the priority of U.S. Provisional Application 60/441,147 filed on January 21, 2002, which is incorporated herein by reference in its entirety.

Field of the Invention

The present invention is directed to automatically designing experiments for a semiconductor tool, running the experiments and creating models based on the results obtained by running the experiments.

Background of the Invention

As microelectronic device feature sizes continue to shrink, it is necessary to have tighter controls to maintain high yields. Such tighter controls begin at a tool level. A conventional tool **100** is schematically shown in FIG. 1. The tool **100** can include an etcher, depositor, polisher or the like. Any combination of these can also be included in the tool. A typical tool is controlled by a tool controller **103** which communicates with a factory controller via a communication port **105**. In particular, the tool controller **103**

may receive process recipes from the factory controller via the communication port **105** and process wafers in accordance with the received recipes.

The tool **100** can be controlled on a run-to-run control basis for various semiconductor manufacturing processes. The run-to-run control reduces unacceptable variations of outputs (i.e., wafers processed by the tool) from targets. In the run-to-run control of such a tool, the process recipe is modified between process runs so as to minimize process drift, shift, and variability.

Creating accurate and precise run-to-run control starts from designing and running experiments on the tool for an eventual modeling of the tool. Designing a set of experiments is called DOE (Design of Experiments). A good DOE establishes the relationship between variables that may have a predictable impact on the processing output a user wishes to control, e.g., one or more film properties such as film thickness, while keeping the required number of experiments low.

Conventionally, a DOE system **107** configured to generate a DOE plan that includes a set of experiments is typically not integrated with the tool **100**. Hence, the experiments of the DOE plan are run on the tool **100** by a user manually setting up the tool **100**. When the experiments of the DOE plan are run, data relating to process recipe parameters and process outcome are collected. The collected data are then used in creating one or more models in a modeling environment **109**.

Conventionally, the modeling environment **109** is also not integrated with the tool **100**. In the modeling environment **109**, the models are created, and the models can be

represented as raw data that reflects the tool, or it can be represented by equations, for example, multiple input-multiple output linear, quadratic and general non-linear equations, which describe the relationship among the variables of the tool **100**.

The DOEs, models and eventual run-to-run control of tools are, conventionally, performed on a lot-to-lot basis. This is because it is difficult to collect the data from different tools, put them together and control experiments at a wafer-to-wafer level. As noted above, the tool **100**, DOE system **107**, and modeling environment **109** are not integrated together. Therefore, once a DOE plan is created, its experiments are run manually on the tool **101** and the resulting data are collected manually. Even if the DOE data are collected electronically, it needs to be reformatted to be used in the modeling environment **109**. This also means that there cannot be any automated coordination between the DOE systems **107** and modeling environment **109**. These shortcomings made the use of the DOEs a difficult process for a user of the tool **100**.

Summary of the Invention

In embodiments of the present invention, a user is allowed to design experiments, i.e., use DOE methodologies, and then automatically execute the experiments on a tool and automatically collect all the data related to the experiment. The automation is achieved by, among other things, integrating the DOE system and modeling system with the tool.

Once the above steps have been completed, the collected data from the DOE run is used in creating one or more models that can be used in generating process recipes to control tools.

Description of the Drawings

The detailed description of the present application showing various distinctive features may be best understood when the detailed description is read in reference to the appended drawings in which:

FIG. 1 is a schematic diagram of an example semiconductor processing tool, a DOE system and a modeling environment;

FIG. 2 is a block diagram/flow chart illustrating high level processes of embodiments of the present invention;

FIG. 3 is an image of a graphical user interface configured to capture factors of a DOE plan by interacting with a user according to embodiments of the present invention;

FIG. 4 is an image of a graphical user interface configured to capture responses of a DOE plan by interacting with a user according to embodiments of the present invention;

FIG. 5 is an image of a graphical user interface configured to capture parameters for an Auto Generate window of a DOE plan by interacting with a user according to embodiments of the present invention;

FIG. 6 is a graph illustrating factorial and central composite designs according to embodiments of the present invention;

FIG. 7 is an image of a graphical user interface illustrating experiment cases of a DOE plan by interacting with a user according to embodiments of the present invention;

FIG. 7A is an image of a graphical user interface illustrating block selections of a DOE plan by interacting with a user according to embodiments of the present invention;

FIG. 7B is an image of a graphical user interface illustrating designed experiments of a DOE plan by interacting with a user according to embodiments of the present invention;

FIG. 8 is an image of a graphical user interface configured to capture a Formula Type of a model creation by interacting with a user according to embodiments of the present invention;

FIG. 9 is an image of a graphical user interface illustrating statistical results according to embodiments of the present invention;

FIG. 10 is a flow chart illustrating various steps in generating a model according to embodiments of the present invention;

FIG. 11 is a flow chart illustrating various steps in designing a DOE plan according to embodiments of the present invention;

FIG. 12 is a block diagram illustrating communication links among a DOE system and a number of tools according to embodiments of the present invention;

FIG. 13 is a block diagram representation of example embodiments of a computer configured to perform embodiments of the present invention; and

FIG. 14 is a diagram illustrating an example of a memory medium embodiments of the present invention, which may be used for storing computer programembodiments of the present invention.

Detailed Description

Various embodiments of the present invention can be implemented in hardware, firmware, software or any combination of them. For the sake of clarity, the descriptions below are provided in terms of software implementations. In particular, the description is provided in the vernacular of the object-oriented programming field. However, the concepts of embodiments of the present invention are not limited to the implementations in the object-oriented programming field.

Now referring to FIG. 2, the figure illustrates a high level a part flow chart and part block diagram of embodiments of the present invention. Embodiments of the present invention include a process that is a combination of various parts such as a DOE **201**, a Model Gen. **205** and a Controller **207**. In particular, the process controller includes software programs necessary to design a DOE plan for a tool (a Processor Controller–DOE) **201**, create a corresponding model (a Processor Controller–Model Gen.) **205** of the tool and a controller (a Processor Controller–Control) **207** to control the tool using

process recipes generated based on the model. All these software programs are integrated with each other (e.g., send/receive data, coordinate actions, etc.)

In particular, a DOE plan is automatically designed and its experiments are performed on a tool **203**. As the experiment of the DOE plan are run, the process controller–DOE **201** automatically collects and stores data. The stored data are then used by the model generator of the process controller–Model Gen. **205** to automatically create a model of the tool **203**. The process controller–Control **207** is capable of automatically generating one or more process recipes for processing one or more wafers on the tool **203**. Each of the above steps is described in detail below.

Embodiments of the present invention are described in terms of two perspectives. The first is described in terms of a set of user interfaces in FIGs. 3-9, the second is described in terms of a set of flow charts in FIGs. 9-10.

Now turning to describe embodiments of the present invention in terms of user interfaces, FIGs. 3-6 illustrate various features in an example user interface window **300** that allows a user to make selections for automatically generating a DOE plan. The window is called a "New DOE Plan" (hereinafter the "DOE window"). Using the DOE window **300**, the user may set factors, responses, cases, blocks and experiments. Each of these is described in detail below.

As shown in FIG. 3, the DOE window **300** includes panels for a "Current State" **301**, "DOE plan name" **303**, and "Resource type" **305**. With respect to the Current State **301**, in the vernacular of the object-oriented programming field, each object in

embodiments of the present invention is created as inactive, including the DOE plan object. The DOE plan object may remain inactive until an Activate button **341** is selected by the user. With respect to the DOE plan name **303**, it is the name of the DOE plan to be created. The user may arbitrarily enter the name. With respect to the Resource type **305**, it is the tool for which the DOE plan is to be created. In this example, the type of the tool is an oxide CMP (Chemical-Mechanical-Planarization) profiler.

In addition to the panels, in FIG. 3, three buttons are shown at the right-hand side: a "New" button **335**, "Open" button **337**, "Delete" button **339**. These buttons allow the user to create a new DOE plan, open an existing DOE plan, and delete an existing DOE plan, respectively.

Five tabs are also illustrated in FIG. 3: Factors **307**, Responses **309**, Cases **311**, Blocks **317**, and Experiments **315**. In FIG. 3, the table shown (headed by "Factor Name") **317** corresponds to when the Factors tab **307** is selected. A factor is a parameter to be adjusted between experiments while running the DOE plan. In other words, factors can be recipe parameters of the tool. The user can select a set of factors from a field of factors. As the example depicted in FIG. 3 shows, the factors can be a baseline time **319**, center time **321**, edge time **323**, etc. These are the parameters relating to an oxide CMP profiler tool. For this set of factors, their units are in milliseconds. The term "LSL" **327** means the lower specification limit, and the term "USL" **329** means the upper specification limit. After selecting the factors, the user can enter the values for the LSL and USL. The user can specify if the factors are to be time-based or constant value.

The time-based parameter is a parameter that has linear output values in function of time. If the parameter is actually not a linear function in time, the collection of the output values can be time scaled to ensure that the collected data appear as a linear function in time. This may remove any non-linearity in the collected data. For the constant valued parameter, the value of the designated parameter is set at a specific value through out the experiments.

In FIG. 4, the table shown (headed by "Response Name") **401** corresponds to when the Responses tab **309** is selected. The responses are the data collected based upon the performance of the tool. For example, as shown in FIG. 4, the data can be collected for a thickness **403** and pad life **405** of the oxide CMP profiler tool.

After the factors and responses have been selected, the user then may select the cases tab **311**. When the cases tab is selected, a cases table will be shown to the user. An example of a cases table **711** is shown in FIG. 7. Although the example cases table **711** in FIG. 7 illustrates ten cases, the table **711** may show no cases when the cases tab is first selected. Subsequently, the cases can be automatically generated by selecting an "Auto Generate" button **701**. When the "Auto Generate" button **701** is selected, an Auto Generate window **501** is shown to the user, as illustrated in FIG. 5. Using the Auto Generate window **501**, the user is allowed to select the DOE structure **503**, DOE method **505** and DOE fraction **519**.

For the model structures, the user is allowed to select one of four (each either with or without a constant term, listed as k_0 in the equation below): 1) Linear without

interactions **507**; 2) Linear with interactions **509**; 3) Quadratic without interactions **511**; and 4) Quadratic with interactions **513**. An example equation of the linear without interactions can be

$$y = k_0 + k_1x_1 + k_2x_2 + k_3x_3$$

where the y represents predicted output values, the values of k_i are chosen such that the value of y is "close" to the value of the response which is being modeled for all the values of x_i in the experiment, and the x's represent values of recipe parameters or other measured parameters. An example equation of the linear with interactions can be:

$$y = k_0 + k_1x_1 + k_2x_2 + k_3x_3 + k_{12}x_1x_2 + k_{23}x_2x_3 + k_{13}x_1x_3$$

One difference between the linear without interactions and linear with interactions structures is that the equation with interactions model structure includes multiplied factors (e.g., the x_1x_2 and x_2x_3 terms). With respect to the quadratic model structures, they are similar to the linear structures except that the quadratic model structures include quadratic terms instead of the linear terms, e.g.,

$$y = k_0 + k_1x_1 + k_2x_2 + k_3x_3 + k_{12}x_1x_2 + k_{23}x_2x_3 + k_{13}x_1x_3 + k_{11}x_1^2 + k_{22}x_2^2 + k_{33}x_3^2$$

After selecting a model structure, a DOE method may be selected. Two example DOE methods are provided: a factorial design **517** and central composite design **515**. The factorial design generates cases for all possible combinations of high and low for all the parameters involved for a full factorial design. A schematic depiction of cases for a full factorial design **601** of a two-parameter set is shown in FIG. 6. More specifically,

once a central point **603** has been selected, the full factorial design generates minimum and maximum combinations from the central point. The central composite design **605** is also graphically illustrated in FIG. 6. These designs are well known in the art.

The central composite design is preferably used for non-linear models, and the factorial model is preferably used for linear type models. However, in embodiments of the present invention, the central composite design can be used for linear models, and the factorial model can be used for non-linear type models. With respect to the DOE fraction **519**, generating and running full factorial or central composite experimental designs can generate statistically redundant data points. Using this characteristic, a fraction of the experiments can be selected (e.g., Full, ½, ¼, etc.) as shown in the DOE faction sub-panel **509**.

Based on the selections made by the user, corresponding cases are automatically generated when the user clicks an "OK" button **521**. An example of resulting set of cases is illustrated in a table headed by "Case" **711** in FIG. 7. In that example, ten cases are listed, one case for each extreme values and middle values of the baseline time, center time, and edge time. (Refer back to FIG. 4 for the extreme values of the factors.)

In embodiments of the present invention, the cases/experiments can also be imported from other DOE systems by using an "Import" button **703**. An example of the other DOE systems can be Design-Expert Software manufactured by Stat-Ease, Inc., located in Minneapolis, MN.

Manual entry of cases and/or manipulations of imported or auto-generated cases are also contemplated within embodiments of the present invention. More specifically, the case values can manually be typed into the table and rows can be added, inserted, or deleted by an "AddRow" button **705**, "InsertRow" button **707**, and "DeleteRow" button **709**, respectively.

Once a list of cases is generated, the list can also be blocked. In a "Blocks" panel, a number of blocks can be selected and specified for each list and corresponding responses. An example block set is illustrated in FIG. 7A. For example, one block can be assigned for three pad life values (e.g., high, mid, low). In such an example setup, thirty (i.e., ten cases times three blocks) experiments would eventually be generated. An example list of designed experiments are illustrated in FIG. 7B.

For a novice user, the above-described parameters can be preset. In particular, the entries made using the graphical interfaces described above can be preset to a set of specific preset values. This allows the steps of designing a DOE plan, running the experiments and collecting the data to all be automated with minimal user intervention.

After the DOE plan is automatically generated, the experiments specified in the DOE plan are run. Because the processor controller–DOE **201** is integrated with the tool **203**, the DOE plan is automatically executed without user interventions. The user can observe as experiments are being performed and respond to any possible alerts (e.g., supply more wafers to the tool to complete the experiments). The experiments are run and data are collected on a wafer-by-wafer basis rather than on a lot-by-lot basis.

The data collected while running the experiments in the DOE plan are automatically formatted such that they can be used by the processor controller–Model Gen. **205** in creating a run-to-run model. In preparation of creating the run-to-run model, the user can select a formula type using, for example, a Formula type window **801** shown in FIG. 8.

More specifically, the Formula type window **801** allows the user to select one of four formula types: linear without interactions **803**; linear with interactions **805**; quadratic without interactions **807**; and quadratic with interactions **809**. The equations for these formulas are similar to the equations describe above in connection with FIG. 5. In additions to the four formulas, the user is allowed to enter a more general time-based or non-time-based linear-in-parameters model structure by selecting a "Use Template" option **813**. Using this option, the user then enters terms **817** and designates whether the terms are time-based in a Time-Base designating field **819**.

In embodiments of the present invention, the collected data can be transformed to create new sets of data. For example, assume the DOE plan collected a set of measurements on a wafer, e.g., 1-25, the first 5 corresponding to a specific region on the wafer, the second 5 corresponding to another region on the wafer, then a transformation can be setup so that data for region 1 as the average of the first 5 points can be created. Subsequently, a model to determine how certain measured variables or recipe parameters could affect this specific controlled output can be created. In another example, taking a logarithm of a certain set of the collected data can create a new set of data.

When a formula is selected, the coefficients for the formula are calculated by the regression method. The method is performed, for example, in the following manner for a multiple input, single output system:

$$\hat{y}(k) = \mathbf{b}^T \varphi(k),$$

where $\hat{y}(k)$ is the (scalar) output for each experiment k of the predicted values,

$\varphi(k)$ is the vector of inputs for each experiment k, and

b is a vector of model coefficients.

The set of data obtained in the DOE is represented as:

$$\mathbf{y} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}, \mathbf{\Phi} = \begin{bmatrix} \varphi^T(1) \\ \vdots \\ \varphi^T(N) \end{bmatrix}$$

where **y** is a vector of the measured outputs from the DOE, and $\mathbf{\Phi}$ is a matrix made up of the vectors of inputs corresponding to the conditions that result in each of the elements of *y(k)*. The model coefficients, **b**, which provides the best fit (e.g., in the least squares sense) is provided by following equation:

$$\mathbf{b} = \left(\mathbf{\Phi}^T \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^T \mathbf{y}$$

An estimate of the covariance matrix for **b** is given through similar calculations. The correct method for estimating the covariance matrix, **P**, is given by:

$$\mathbf{P} = \sigma\left[\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^{T}(k)\right]^{-1}$$

where σ is the prediction error variance defined as:

$$\sigma = \frac{1}{N}\sum_{k=1}^{N}\varepsilon^{2}(k)$$

and

$$\varepsilon^{2}(k) = \left(y(k) - \hat{y}(k)\right)^{2}$$

with $\hat{y}(k)$ being the prediction at time k as defined above.

Once the coefficients are calculated as described above, the model is created based on the formula selected in FIG. 8 and the coefficients. An example of the accuracy of the predictions of an example model **901** can be statistically illustrated in FIG. 9. The DOE results in FIG. 9 show predicted values **903**, actual values **905** and residual values **907**.

FIG. 9 also includes an ANOVA (analysis of variance) window **909** configured to show statistical values such as the R^2 value **911**, adjusted R^2 value **913** and P value **915**. These statistical values are calculated as known in the art.

When the selected model yields unsatisfactory results, a different model can be selected. However, if no model yields satisfactory results one or more of the following approaches can be taken depending upon the source of the modeling issue: 1) start over

by designing a new DOE plan to execute and collect data; 2) run a new set of experiments with a new DOE augmenting the existing data; 3) import previously run data; and/or 4) import data from another DOE system (e.g., an external system such as the Design-Expert described above). With respect to the imported data, the modeling and analyses can be performed using only the imported data, the data collected data, or any combination of them. This allows the imported data to augment the collected data. Once a satisfactory model is created, the model is then used in controlling the tool. The actual control is performed by process controller-Control **207**.

Now turning to describe embodiments of the present invention using flow charts, FIGs. 10 and 11 illustrate example steps in the designing and running of a DOE plan and creating a run-to-run model. In particular, FIG. 10 describes the running of a DOE plan and FIG. 11 describes the designing of a DOE plan.

First turning to FIG. 10, the user decides whether to use a pre-configured DOE plan and association without changes. The association means that the factors and responses in the DOE plan are associated with the actual data, which are collected or sent to the tool. If the user chooses to design a new DOE plan, the example steps shown in FIG. 11 are executed. If the user chooses to use a pre-configured DOE plan and association, the following steps are executed.

In step **1005**, the user activates the DOE plan and association (either newly created, selected from the pre-configured ones, and/or imported from an external source). In step **1007**, the user can configure wafers on the tool and run the activated DOE and

association. In embodiments of the present invention a graphical user interface to display runtime data and capability can also be provided as described above in connection with the example graphical user interfaces. At this point, the actual run of the DOE and association is automated in embodiments of the present invention.

In step **1009**, the data obtained as the result of running the DOE and association are stored. In embodiments of the present invention, the stored data can be edited or saved under a user specified name. A graphical user interface can be provided for this purpose.

In step **1011**, the user is asked to decide either to use a pre-configured run-to-run (R2R) model or an association. If a pre-configured run-to-run model and run-to-run association is to be used, the next step is **1019**. Otherwise, steps **1013**, **1015** and **1017** are performed.

In step **1013**, if a run-to-run model does not exists, a run-to-run model based on the DOE plan is created. In particular, the factors are automatically converted to variables such as manipulated parameters and responses are mapped to variables such as measured variables, raw data parameters, transformation inputs or controlled outputs.

In step **1015**, the user may define transformations. Example transformations were described above in connection with FIG. 8.

In step **1017**, controlled output models with place holders for coefficients are defined.

Now turning back to step **1019**, which would have been executed had there been a pre-configured run-to-run model and association, the DOE results (saved in step **1009**) are retrieved and a controlled output model is identified.

In step **1020**, a determination is made as to whether the model is adequate for the designated application. If it is not adequate, then the steps from **1001**, **1007**, **1015** or **1017** can be repeated. For instance, when a DOE plan is available, steps from step 1007 can be repeated, and, when a DOE plan and R2R model are available, steps from step 1015 can be repeated.

In step **1021**, the user is to determine whether to use a pre-configured run-to-run association. If not, then a run-to-run association is performed using "Save As" for association or through a graphical user interface.

Once a run-to-run association is selected (either in step **1021** or step **1023**), the run-to-run model, run-to-run association, and manipulated process recipes may all be activated and run in a stand-alone mode or export mode so that an import can be performed on a module controller **1207** or another APC system. The module controller and APC system are described below in connection with FIG. 12.As noted above, FIG. 11 illustrates the steps of designing a DOE plan. First, in step **1101**, the user configures a DOE plan by selecting factors, e.g., factor USL and LSL, and responses.

In step **1103**, the user determines whether to enter DOE cases manually. If the user decides to enter the cases manually, then the user makes such entries in step **1105** and skips to step **1119**.

In step **1107**, the user is allowed to import a DOE plan. If a DOE plan is to be imported, then the names in the import file are checked to ensure that they match names in the DOE plan in step **1122**. After such a check, the imported file can be edited in step **1117**.

If the user does not import a DOE plan, a candidate model structure is selected from the choices of linear, linear with interaction, quadratic, and quadratic with interaction in step **1109**. Then, a method is selected (e.g., factorial or CCD) in step **1111**, and a DOE fraction is selected in step **1113**.

After the selections have been made DOE cases are generated automatically in step **1115**. The generated cases may be edited by the user in steps **1117**.

In step **1119**, associations are either created or copied for all factors and responses in the DOE plan. The user then selects activated, manipulated process recipes for the DOE. As this point, the control of the steps is reverted back to step **1005** in FIG. 10.

As noted above in connection with FIG. 2, the process controller-DOE **201** is integrated with the process controller-Model Gen. **205**. For instance, if the quadratic with interactions model is selected, the corresponding model (e.g., the quadratic with interactions formula) may require a larger set of experiments in order to collect the necessary amount of information in order to create that model. The linear models, either with or without interactions, would require a much fewer set of experiments. Accordingly, depending upon which model is to be selected, the DOE plan is generated with a larger or smaller set of experiments.

In FIG. 12, an example factory layout of embodiments of the present invention is illustrated. The example layout includes two tools **1201, 1203** connected to an APC (Application Process Control) console **1205**, module controller **1207** and one or more operator consoles **1209**. The tools are connected to the consoles and controllers via two communication ports, FA #1 and FA#2.

In each tool, the APC **1205** communicates with a specific tool and the APC can be accessed by the APC console **1205** or the operator console **1205** remotely. The APC connects to the tool via the connection tools. With respect to the APC, at least some of its various features are also described in U.S. Patent No. _______________, matured from U.S. Non-Provisional Application No. 10/174,377, entitled as "FEEDBACK CONTROL OF SUB-ATMOSPHERIC CHEMICAL VAPOR DEPOSITION PROCESSES," filed on June 18, 2002, which is incorporated herein by reference in its entirety.

It should be understood that the various functions, industries, mechanisms, etc. mentioned in the examples above are merely by way of illustration, and that embodiments of the present invention contemplate use in any number of other types and variations of applications.

An example embodiment of the computer in which at least some embodiments of the present invention operates is described below in connection with FIGs. 13-14. FIG. 13 illustrates a block diagram of one example of the internal hardware **1313** of a computer configured to perform embodiments of the present invention. A bus **1356**

serves as the main information highway interconnecting various components therein. CPU **1358** is the central processing unit of the internal hardware **1313**, performing calculations and logic operations required to execute embodiments of the present invention as well as other programs. Read only memory (ROM) **1360** and random access memory (RAM) **1362** constitute the main memory. Disk controller **1364** interfaces one or more disk drives to the system bus **1356**. These disk drives are, for example, floppy disk drives **1370**, or CD ROM or DVD (digital video disks) drives **1366**, or internal or external hard drives **1368**. These various disk drives and disk controllers are optional devices.

A display interface **1372** interfaces display **1348** and permits information from the bus **1356** to be displayed on display **1348**. Communications with external devices, such as the other components of the system described above, occur utilizing, for example, communication port **1374**. Optical fibers and/or electrical cables and/or conductors and/or optical communication (e.g., infrared, and the like) and/or wireless communication (e.g., radio frequency (RF), and the like) can be used as the transport medium between the external devices and communication port **1374**. Peripheral interface **1354** interfaces the keyboard **1350** and mouse **1352**, permitting input data to be transmitted to bus **1356**. In addition to these components, the internal hardware **1313** also optionally includes an infrared transmitter and/or infrared receiver. Infrared transmitters are optionally utilized when the computer system is used in conjunction with one or more of the processing components/stations/modules that transmit/receive data via infrared signal transmission.

Instead of utilizing an infrared transmitter or infrared receiver, the computer system may also optionally use a low power radio transmitter **1380** and/or a low power radio receiver **1382**. The low power radio transmitter transmits the signal for reception by components of the production process, and receives signals from the components via the low power radio receiver. The low power radio transmitter and/or receiver are standard devices in the industry.

Although the computer in FIG. 13 is illustrated having a single processor, a single hard disk drive and a single local memory, the analyzer is optionally suitably equipped with any multitude or combination of processors or storage devices. For example, the computer may be replaced by, or combined with, any suitable processing system operative in accordance with the principles of embodiments of the present invention, including sophisticated calculators, and hand-held, laptop/notebook, mini, mainframe and super computers, as well as processing system network combinations of the same.

FIG. 14 is an illustration of an example computer readable memory medium **1484** utilizable for storing computer readable code or instructions. As one example, medium **1484** may be used with disk drives illustrated in FIG. 13. Typically, memory media such as a CD ROM, a digital video disk or a floppy disk will contain, for example, a multi-byte locale for a single byte language and the program information for controlling the modeling environment, the DOE system, the process control, etc. to enable the computer to perform the functions described herein. Alternatively, ROM **1360** and/or RAM **1362** illustrated in FIG. 10 can also be used to store the program information that is used to

instruct the central processing unit **1358** to perform the operations associated with various automated processes of the present invention. Other examples of suitable computer readable media for storing information include magnetic, electronic, or optical (including holographic) storage, some combination thereof, etc.

In general, it should be emphasized that the various components of embodiments of the present invention can be implemented in hardware, software or a combination thereof. In such embodiments, the various components and steps would be implemented in hardware and/or software to perform the functions of embodiments of the present invention. Any presently available or future developed computer software language and/or hardware components can be employed in such embodiments of the present invention. For example, at least some of the functionality mentioned above could be implemented using Visual Basic, C, C++, or any assembly language appropriate in view of the processor(s) being used. It could also be written in an interpretive environment such as Java and transported to multiple destinations to various users.

The many features and advantages of embodiments of the present invention are apparent from the detailed specification, and thus, it is intended by the appended claims to cover all such features and advantages of the invention which fall within the true spirit and scope of the invention. Further, since numerous modifications and variations will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and operation illustrated and described, and accordingly, all suitable

modifications and equivalents may be resorted to, falling within the scope of the invention.